Troubleshooting em rede básica

www.4linux.com.br

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Troubleshooting em rede básica

1.1. Objetivos

- Inspecionar e resolver problemas na rede;
- Tabela de roteamento.

1.2. Mãos a obra

A configuração básica da rede permite que os hosts (computadores) , comuniquem entre si trocando informações e acessando serviços de um servidor. A perda de conexão com a LAN ou a WAN pode ser resolvida, verificando arquivos de configuração e utilizando comandos para inspecionar e resolver determinados problemas. Vamos á pratica!

Verifique o IP de sua interface de rede usando o comando ifconfig.

 # ifconfig

```
debian:~# ifconfig
eth0      Link encap:Ethernet  Endereço de HW 08:00:27:08:d2:3f
          inet end.: 192.168.200.10  Bcast:192.168.200.255  Masc:255.255.255.0
          endereço inet6: fe80::a00:27ff:fe08:d23f/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:3663 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3665 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:846885 (827.0 KiB)  TX bytes:1157673 (1.1 MiB)

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128 Escopo:Máquina
          UP LOOPBACKRUNNING  MTU:16436  Métrica:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
          RX bytes:1100 (1.0 KiB)  TX bytes:1100 (1.0 KiB)
```

O comando ifconfig exibe todas as interfaces de rede, exibindo IP, mascara de rede, broadcast, mac address, mtu, etc. Você trazer informações sobre a interface da rede usando também o comando ip. Veja alguns comandos:

 # ip link show eth0

```
debian:~# ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:08:d2:3f brd ff:ff:ff:ff:ff:ff
```

Exibir informações de endereçamento IP

 # ip link show eth0

```
debian:~# ip address show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:08:d2:3f brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.10/24 brd 192.168.200.255 scope global eth0
    inet6 fe80::a00:27ff:fe08:d23f/64 scope link
       valid_lft forever preferred_lft forever
```

Para editar as configurações de endereçamento IP, essas seja lidas na inicialização do sistema, edite o arquivo /etc/network/interfaces:

 # vim /etc/network/interfaces

```
 1 # This file describes the network interfaces available on your system
 2 # and how to activate them. For more information, see interfaces(5).
 3
 4 # The loopback network interface
 5 auto lo
 6 iface lo inet loopback
 7
 8 # The primary network interface
 9 allow-hotplug eth0
10 iface eth0 inet static
11         address 192.168.200.10
12         netmask 255.255.255.0
13         network 192.168.200.0
14         broadcast 192.168.200.255
15         gateway 192.168.200.254
```

No Debian você pode usar 2 comandos para desativar e ativar a configuração de um interface de rede. Vamos a prática:

Para desativar a placa eth0:

 # ifdown eth0

Para ativar a placa eth0:

```
# ifup eth0
```

Os comandos só vão funcionar se a interface estiver configurada no arquivo /etc/network/interfaces

Dentro da rede, para que os dados possam chegar ao seu destino, é preciso que haja uma tabela de roteamento na maquina de origem. A função da tabela de rotas é determinar o destino de cada pacote na hora que sai da interface da rede. Há quatro esquema básicos de tabela de rotas:

Mínima → Para redes isoladas, feita quando a interface é iniciada;

Estática → Para redes com um ou mais gateways;

Dinâmica → Para redes maiores, rotas e gateways fornecidos via protocolos;

Estática/Dinâmica → Tabela de rotas que contem informações estáticas, que encaminham pacotes na rede local, que aponta para gateways que trabalham com roteamento dinâmico.

A principio a sua rota padrão é sua própria interface de rede, mas se você quiser comunicação com outras redes sendo elas LAN ou WAN, é preciso definir essas rotas através de alguns comandos. Vamos a pratica.

Use o comando route -n para exibir a tabela de roteamento:

```
# route -n
```

```
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções Métrica Ref   Uso Iface
192.168.200.0   0.0.0.0         255.255.255.0   U      0       0       0 eth0
0.0.0.0         192.168.200.254 0.0.0.0         UG     0       0       0 eth0
```

Para identificar problemas no envio de pacotes, mesmo que você tenha uma rota padrão configurada, veja a descrição das letras exibidas na coluna “Opções”

U → Rota configurada corretamente;
H → O alvo é um hosta tradicional;
G → Gateway padrão;
R → Restabelecer rota por roteamento dinamico;
D → Rota estabelecida dinamicamente;
M → Rota modifica ou redicionada;
! → Rota rejeitada.

Uma outra maneira de exibir informações de roteamento é através do comando ip.

 # ip route list

```
debian:~# ip route list
192.168.200.0/24 dev eth0  proto kernel  scope link  src 192.168.200.10
default via 192.168.200.254 dev eth0
```

Como faço para excluir ou criar novas rotas?

Através do comando route você pode usar as opções add e del para criar ou excluir rotas. Vamos a prática:

Exclua a rota padrão:

 # route del default

Adicione uma nova rota padrão:

 # route add default gw 192.168.200.254

E quando os computadores estiverem em uma outra classe de rede?

Um exemplo simples é quando sua maquina estiver na classe C (192.168.0) e você precisa acessar um servidor qua esta na classe B (172.16.3). Neste caso é preciso adicionar uma rota para outra rede. Veja o exemplo:

Primeiro vamos tentar a comunicação com o servidor:

 # ping -c4 172.16.3.50

```
PING 172.16.3.50 (172.16.3.50) 56(84) bytes of data.
From 192.168.200.10 icmp_seq=2 Destination Host Unreachable
From 192.168.200.10 icmp_seq=3 Destination Host Unreachable
From 192.168.200.10 icmp_seq=4 Destination Host Unreachable

--- 172.16.3.50 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2999ms
, pipe 3
```

Veja a sua tabela de roteamento atual:

 # route -n

```
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções Métrica Ref    Uso Iface
192.168.200.0   0.0.0.0         255.255.255.0   U      0      0        0 eth0
0.0.0.0         192.168.200.254 0.0.0.0         UG     0      0        0 eth0
```

Adicione uma rota para rede de classe B:

 # route add -net 172.16.0.0 netmask 255.255.0.0 dev eth0

Veja a sua tabela de roteamento atual e tente pingar novamente!

Mesmo assim ainda não tenho comunicação!!!

Isso acontece porque a outra maquina com IP de classe B deve criar um rota apontando para sua rede de classe C. Na outra maquina use o comando:

 # route add -net 192.168.200.0 netmask 255.255.0.0 dev eth0

No Linux você pode apenas com um comando exibir conexões de rede, tabelas de roteamento, estatísticas de interface e conexões mascadas. O comando netstat é muito útil devido a sua quantidade de funções. Vamos a prática:

Listar portas abertas TCP e UDP

 # netstat -l --inet

```
Conexões Internet Ativas (sem os servidores)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto         Estado
tcp        0      0 debian.empresa.c:domain *:*                     OUÇA
tcp        0      0 localhost.locald:domain *:*                     OUÇA
tcp        0      0 *:ssh                   *:*                     OUÇA
tcp        0      0 localhost.localdoma:ipp *:*                     OUÇA
tcp        0      0 localhost.localdoma:953 *:*                     OUÇA
udp        0      0 debian.empre:netbios-ns *:*
udp        0      0 *:netbios-ns            *:*
udp        0      0 debian.empr:netbios-dgm *:*
udp        0      0 *:netbios-dgm           *:*
udp        0      0 debian.empresa.c:domain *:*
udp        0      0 localhost.locald:domain *:*
udp        0      0 *:ipp                   *:*
```

Listar tabela de roteamento com netstat

 # netstat -r

```
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções   MSS Janela  irtt Iface
localnet        *               255.255.255.0   U          0 0          0 eth0
172.16.0.0      *               255.255.0.0     U          0 0          0 eth0
default         192.168.200.254 0.0.0.0         UG         0 0          0 eth0
```

Varredura de portas

Além de verificar IP, mascaras, rotas portas TCP e UDP, é possível usar uma ferramenta que permite uma análise mais completa. Com o nmap é possível por exemplo descobrir em uma maquina, o numero da porta um serviço, mesmo estando fora do padrão e com um numero muito alto. Antes vamos instalar:

Para instalar no Debian:

 # aptitude install nmap

Para instalar no RedHat:

 # yum install nmap

Para instalar no OpenSuse:

 # zypper install nmap

Para testar use o nmap para descobrir o numero da porta do serviço SSH da maquina:

 # nmap -sV -O 172.16.3

```
Starting Nmap 4.62 ( http://nmap.org ) at 2010-06-19 01:39 BRT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns_servers
Interesting ports on 172.16.3.50:
Not shown: 1713 closed ports
PORT     STATE SERVICE VERSION
111/tcp open  rpcbind
696/tcp open  rpcbind
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.24
Uptime: 0.267 days (since Fri Jun 18 19:15:40 2010)
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at http:
//nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.110 seconds
```

Para você exibir portas acima de 1024 que são consideradas portas altas, use a opção -p 1024-65535 assim sera exibidos serviços dentro deste range.

 # nmap -sV -O -p 1024-65535 172.16.3

```
PORT      STATE SERVICE VERSION
111/tcp   open  rpcbind
696/tcp   open  rpcbind
63600/tcp open  ssh     OpenSSH 4.3 (protocol 2.0)
```

Opções usadas no comando nmap:

-sV – Exibe informações com nomes de serviços e versões;

-O – Exibe o sistema operacional do host pesquisado;

-p – Define um range de portas para escanear.

Verificando o percurso dos dados

Servidores que estão fora da rede, podem apresentar alguma perda de dados e a maneira mais simples de você descobrir, é usando o comando traceroute.

O comando rastreia o percurso d um pacote desde a origem até seu destino. O traceroute utiliza uma tecnica com o camp Ttl do protocolo IP, assim forçando uma resposta ICMP TIME_EXCEEDED de cada gateway ao longo do caminho. Vamos a prática.

 # traceroute 72.14.209.104

```
traceroute to 72.14.209.104 (72.14.209.104), 30 hops max, 40 byte packets
 1  192.168.200.254 (192.168.200.254)  0.179 ms  0.113 ms  0.113 ms
 2  10.11.0.1 (10.11.0.1)  6.825 ms  12.605 ms  12.489 ms
 3   (187.21.64.1)  12.615 ms  12.493 ms  12.371 ms
 4   (187.21.64.100)  12.654 ms  12.532 ms  12.921 ms
 5  bd048081.virtua.com.br (189.4.128.129)  12.799 ms  12.677 ms  12.554 ms
 6  c9060711.virtua.com.br (201.6.7.17)  13.630 ms  9.578 ms  12.745 ms
 7  c906071e.virtua.com.br (201.6.7.30)  12.622 ms  9.340 ms  19.117 ms
 8  spogblrtd01.virtua.com.br (201.6.0.13)  28.860 ms  28.744 ms  28.622 ms
 9  c90601e6.virtua.com.br (201.6.1.230)  19.361 ms  19.244 ms  19.509 ms
10  209.85.249.232 (209.85.249.232)  18.988 ms 209.85.250.246 (209.85.250.246)
19.227 ms 209.85.249.232 (209.85.249.232)  18.705 ms
11  209.85.249.47 (209.85.249.47)  139.412 ms  139.301 ms  131.738 ms
12  209.85.254.252 (209.85.254.252)  158.128 ms 216.239.48.192 (216.239.48.192)
```

Capítulo 2

Gerenciando

2.1. Objetivos

- Numeração correta das interfaces de rede.

2.2. Troubleshooting

Como posso ter a numeração correta da placas de rede?

Se você troca muito as placas de rede em seu computador, e nas maquinas virtuais utiliza instalações já prontas do Linux, vai perceber que as interfaces de rede trocam de numero. Como resolver isso?

Vamos a um exemplo de uma maquina virtual. Use o comando ifconfig -a

```
eth1      Link encap:Ethernet  Endereço de HW 08:00:27:e7:d4:02
          BROADCASTMULTICAST  MTU:1500  Métrica:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128 Escopo:Máquina
          UP LOOPBACKRUNNING  MTU:16436  Métrica:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
          RX bytes:100 (100.0 B)  TX bytes:100 (100.0 B)
```

Veja que temos a interface eth1 sem IP, mais na verdade a única que você gostaria de exibir é a eth0. Para isso edite o arquivo /etc/udev/rules.d/70-persistent-net.rules. Dependendo da distribuição o numero 70 muda.

\# *vim* /etc/udev/rules.d/70-persistent-net.rules

```
 1 # This file was automatically generated by the /lib/udev/write_net_rules
 2 # program run by the persistent-net-generator.rules rules file.
 3 #
 4 # You can modify it, as long as you keep each rule on a single line.
 5
 6 # PCI device 0x8086:0x100e (e1000)
 7 SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:73:
   fb:cc", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
 8
 9 # PCI device 0x8086:0x100f (e1000)
10 SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:e7:
   d4:02", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

Agora apague as linhas 6 e 7 e altere eth1 que sobrou para eth0

Reinicie a maquina e sua eth0 estará de volta!!!